LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Programming MATLAB Symbolic Math Toolbox for Speed: Experience from the GenSSI Version 2 Project

Thomas Ligon

April 26, 2018

Start

<Ligon>

Bioinformatics, 00(0), 2018, 1–3
doi: 10.1093/bioinformatics/btx735
Advance Access Publication Date: 30 November 2017
Applications Note

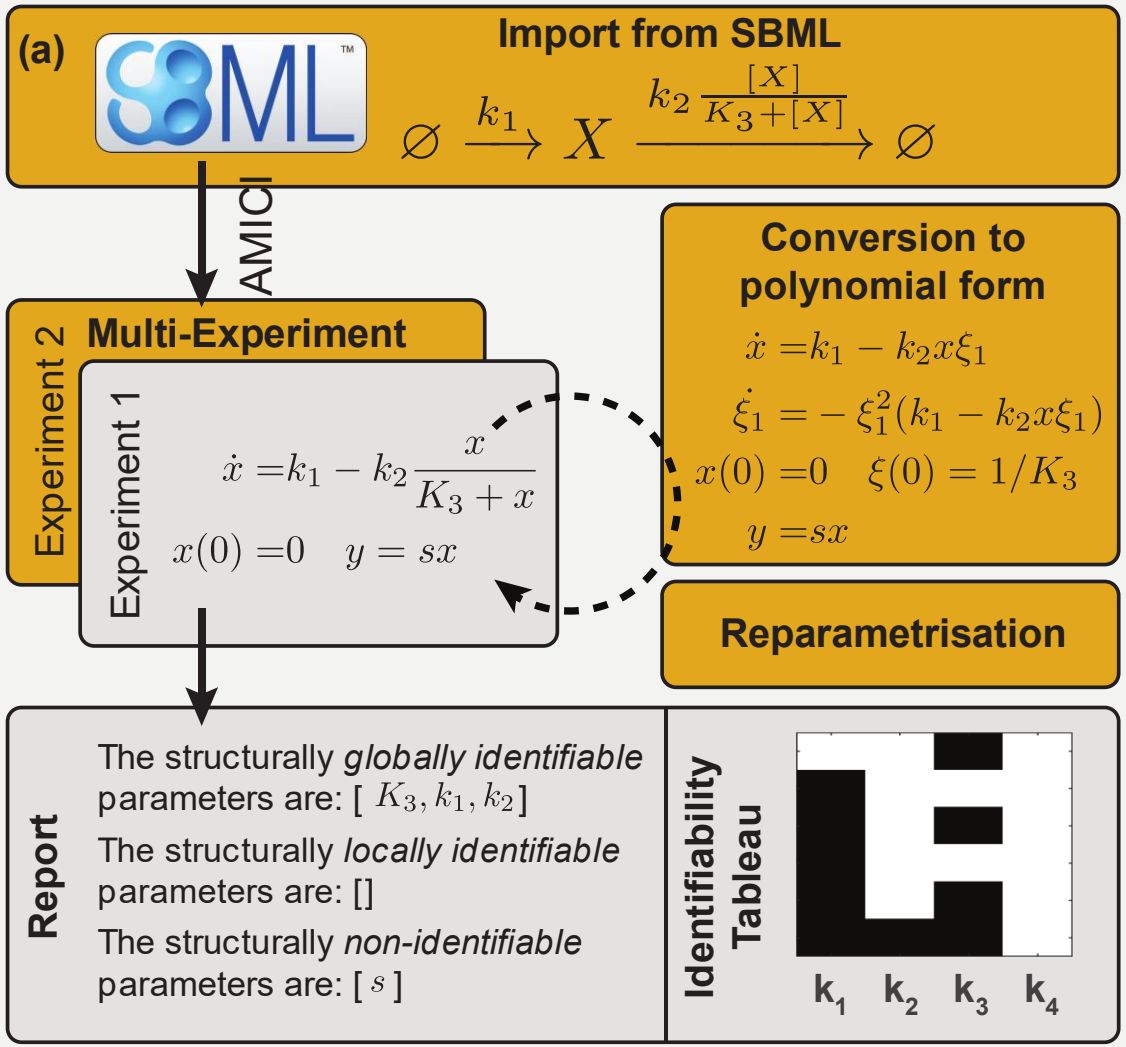
Systems biology

GenSSI 2.0: multi-experiment structural identifiability analysis of SBML models

Thomas S. Ligon[1,*], Fabian Fröhlich[2,3], Oana T. Chiş[4], Julio R. Banga[5], Eva Balsa-Canto[5] and Jan Hasenauer[2,3,*]

[1]Faculty of Physics and Center for NanoScience (CeNS), Ludwig-Maximilians-Universität, 80539 München, Germany, [2]Institute of Computational Biology, Helmholtz Zentrum München, 85764 München, Germany, [3]Center of Mathematics, Technische Universität München, 85748 München, Germany, [4]Technological Institute for Industrial Mathematics, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain and [5](Bio)Process Engineering Group, Spanish National Research Council, IIM-CSIC, 36208 Vigo, Spain

GenSSI 2.0


(a)
SBML
Import from SBML
$\varnothing \xrightarrow{k_1} X \xrightarrow{k_2 \frac{[X]}{K_3+[X]}} \varnothing$
AMICI
Experiment 2
Multi-Experiment
Experiment 1
$\dot{x} = k_1 - k_2 \frac{x}{K_3 + x}$
$x(0) = 0 \quad y = sx$
Conversion to polynomial form
$\dot{x} = k_1 - k_2 x \xi_1$
$\dot{\xi}_1 = -\xi_1^2 (k_1 - k_2 x \xi_1)$
$x(0) = 0 \quad \xi(0) = 1/K_3$
$y = sx$
Reparametrisation
Report
The structurally *globally identifiable* parameters are: [K_3, k_1, k_2]
The structurally *locally identifiable* parameters are: []
The structurally *non-identifiable* parameters are: [s]
Identifiability Tableau
k1 k2 k3 k4

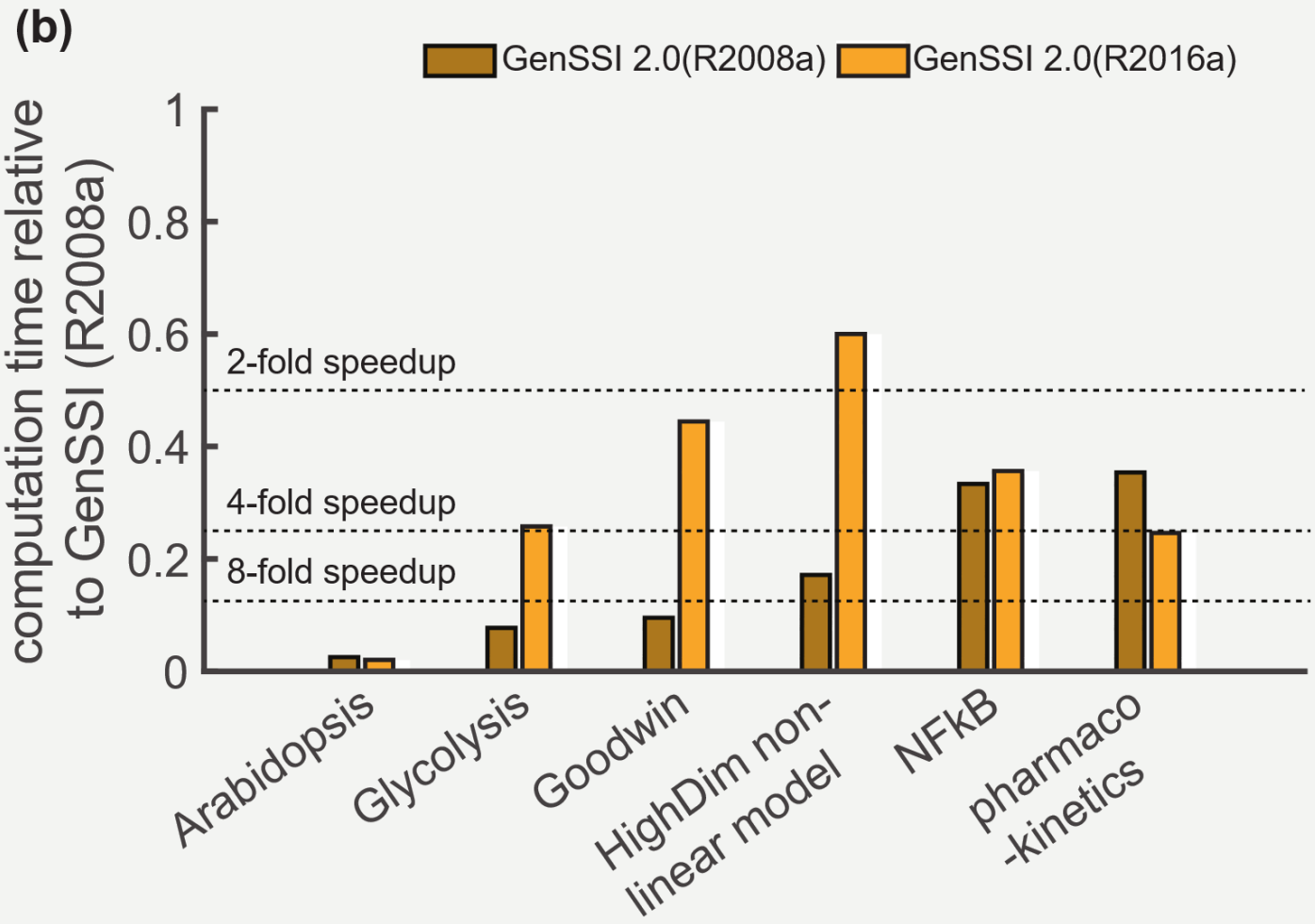
GenSSI Version 1

- GenSSI written in MATLAB with the Maple toolbox
- In 2008, Mathworks acquired MuPAD
- MuPAD developed by University of Paderborn and sold to SciFace
- MuPAD became MATLAB Symbolic Math Toolbox, replacing Maple

GenSSI Version 2

- Convert Code to run in newer version of MATLAB
- Calculation of Jacobian matrix required change
- Many performance problems required change
- New functions, such as multi-experiment model created
- Performance (R2016a) better than Version 1 (R2008a)

GenSSI 2.0 Performance



Jacobian of a Matrix

Maple can calculate Jacobian of a matrix

MATLAB only calculates Jacobian of a vector

Possible solution:

```
df = sym(zeros([size(f,1),size(f,2),length(v)]))
for iRow = 1:size(f,1)
   df(iRow,:,:) = jacobian(f(iRow,:),v)
end
```

This does not work for GenSSI, because

Maple and MATLAB order the result differently

Jacobian of a Matrix

```
% Original code:
rank(LDer); % LDer is matrix of Lie derivatives
JacParamC=jacobian(LDer, Par); % Par is vector of parameters

% Final code:
% calculate 2D jacobian the way Maple does it
JacParam = jacobian(reshape(LDer,[numel(LDer),1]),model.sym.Par);
```

That works, but then we had big performance problems!

Previous Experience

Memory management can be very dangerous

Example of a very efficient system:

- Call the operating system rarely…
- … and manage that buffer internally
- Allocate many pieces of memory (internally)…
- … and manipulate pointers instead of data

But that's low-level code, and we are using MATLAB

Matrices in MATLAB

Numeric matrices have a fixed size

- Number of cells times size of (double precision) number
- Pre-allocation is effective

Symbolic matrices have a variable size

- Cells can be long expressions or polynomials
- Pre-allocation is still not enough

Mathworks Recommends

Mathworks highly recommends Pre-allocation and Vectorization

```
1 rowN = zeros(numCols,1); pre-allocate rowN
2 for iCol = 1:numCols
3    rowN(iCol) = matrix(n,iCol) % elementwise (loop)
4 end
% alternate code
5 rowN = matrix(n,:) % vectorized
```

Without line 1, line 3 changes size of rowN

Line 5 puts all the logic inside of MATLAB

Performance numbers later (in GenSSI code)

Remove Zero Rows - Old

```
1 remove_Lie_index=[];
2 for iRow=JacParamx:-1:1
3     if JacParam(iRow,:)==0;
4         JacParam(iRow,:)=[];
5         remove_Lie_index=[remove_Lie_index iRow];
6     end
7 end
```

Line 4 changes the size of JacParam

- Causes memory management

Line 5 also changes size

Remove Zero Rows - New

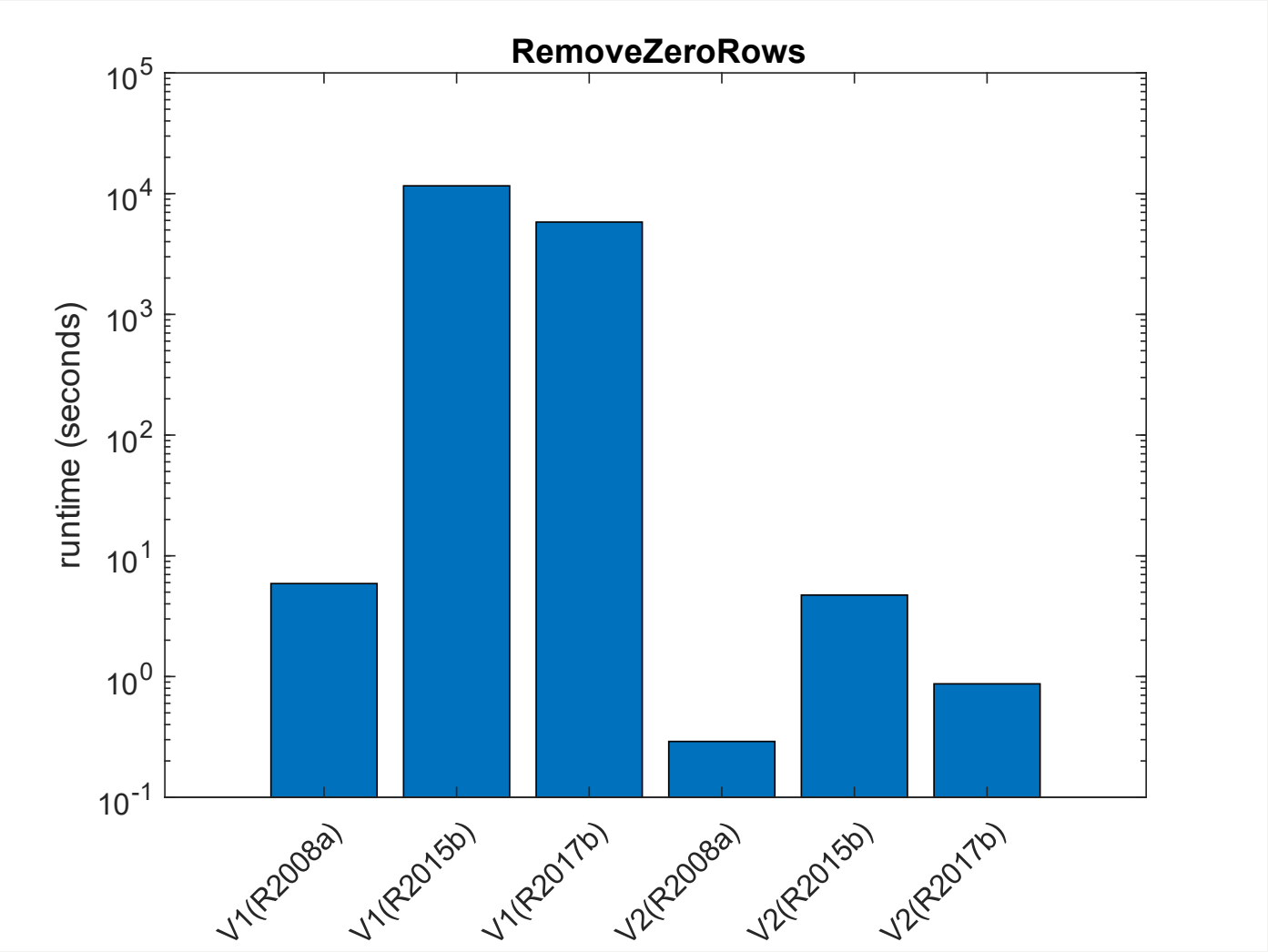
```
1 [JacParam,tilde,useful_Lie_index] = genssiRemoveZeroRows(JacParam);
3 results.useful_Lie_index = useful_Lie_index;

4 function [matrixOut,keepBoolean,keepIndex]=genssiRemoveZeroRows(matrixIn)
5     keepBoolean=any(matrixIn~=0,2);
6     matrixOut=matrixIn(keepBoolean,:);
7     keepIndex=find(keepBoolean)';
8 end
```

Line 5 contains all logic, no "for" loop, no "if"

- "any" creates a Boolean array
- Line 6 returns rows with 1 in keepBoolean

Remove Zero Rows - Results

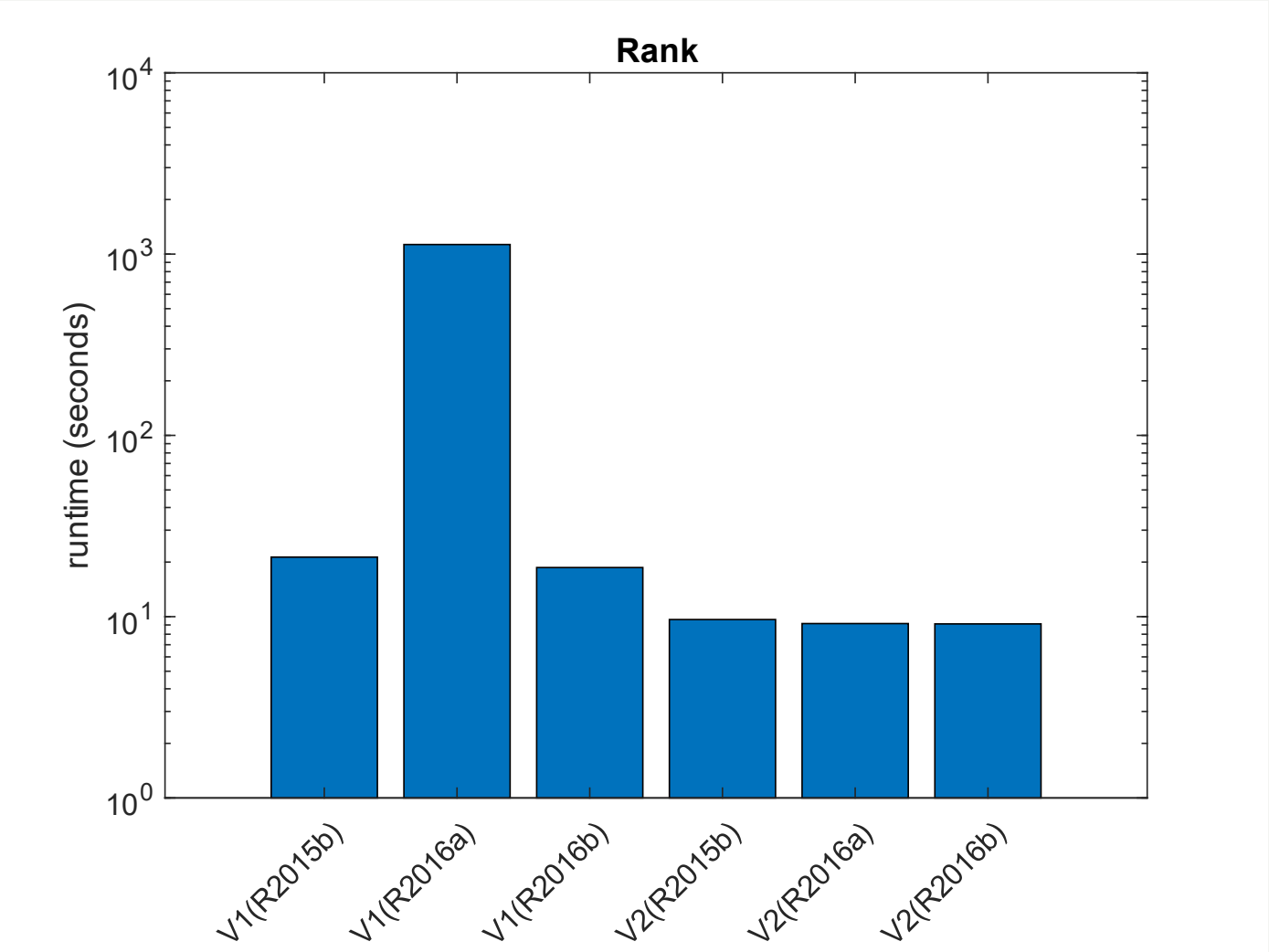


Test Performance of Rank

```
1 load('JacParamC1.mat’);
2 A=JacParamC1;
3 tic;
4 R1=rank(A); % MATLAB Rank
5 disp(['rank1=',num2str(R1),', time=',num2str(toc)]);

6 tic;
7 R2=feval(symengine,'linalg::rank',A); % MuPAD Rank
8 disp(['rank2=',char(R2),', time=',num2str(toc)]);
9 disp('end');
```

MuPAD Rank – Results



Symbolic to Numeric - Code

Convert Jacobian (symbolic) to Tableau (Boolean)

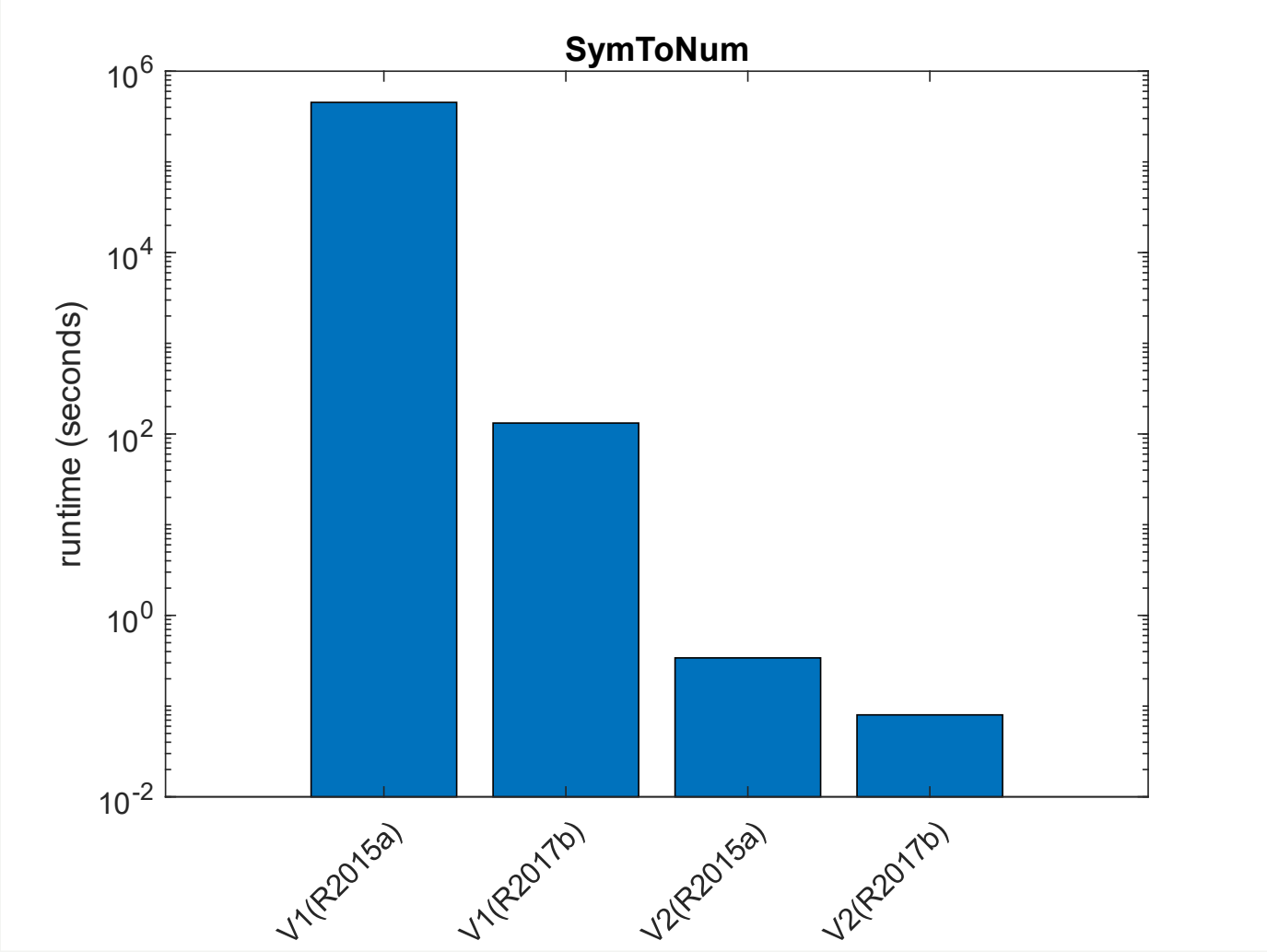
Old code (vectorized single line)

```
% JacParam01=zeros(sizeJacParam);
JacParam01=double(JacParam~=0);
```

New code

```
JacParam01=zeros(size(JacParam));
JacParam01(find(JacParam))=1;
```

Symbolic to Numeric - Results



Remaining Limitations

MATLAB **solve**

- In some cases, it cannot find a solution
- Warning: Explicit solution could not be found
- Limits information returned

MATLAB **jacobian**, **rank**, **solve**

- can be slow
- support for parallel processing would help

Memory

- every new derivative increases the size of the jabobian by a factor of Npar (number of parameters)
- Memory usage grows exponentially

Summary

Beware of things that require memory management

Pre-allocation is good

Vectorization is good

Special functions including “any” and “find” are good

- but might require a version check

Thanks to Mathworks for support cases

- provided quick help and workarounds
- fixed bugs and improved code in future versions

General recommendations

- Support: Always provide a clear and concise test program and description.
- Change requests: Include a business case.

GenSSI Teams

Thomas S. Ligon[1]

Oana-Teodora Chiş[4]

Fabian Fröhlich[2,3]

Eva Balsa-Canto[5]

Jan Hasenauer[2,3]

Julio R. Banga[5]

[1]Faculty of Physics, Ludwig-Maximilians-Universität, München, Germany,

[2]Institute of Computational Biology, Helmholtz Zentrum München, Germany,

[3]Center of Mathematics, Technische Universität München, München, Germany,

[4]Technological Institute for Industrial Mathematics, Campus Vida, Santiago de Compostela, Spain,

[5](Bio)Process Engineering Group, Spanish National Research Council, IIM-CSIC, Vigo, Spain

End

<Ligoff>

Appendix

Abstract

Dynamical systems, especially in systems biology, are often modeled by systems of ordinary differential equations (ODEs) and we want to know if it is possible to infer the parameters of these equations (e.g. reaction rates) from experimental data, in a process called parameter estimation or “the inverse problem”. If this is possible in principle, i.e. based on optimal data, the parameters are called “structurally identifiable”. GenSSI (Generating Series for testing Structural Identifiability) uses generating series of Lie derivatives to analyze the structural identifiability of parameters of a set of ODEs based on arbitrary analytical functions. We converted GenSSI from version 1, which uses the Maple toolbox and runs on MATLAB version R2008a and older, to version 2, which uses the MATLAB Symbolic Math toolbox (based on MuPAD) and runs on MATLAB version R2008b and newer. As part of this, we corrected some very significant performance issues with the Symbolic Math toolbox, ultimately achieving better performance than the original version. This talk addresses those performance aspects.

Tools for identifiability

Table S1. Implementation and availability of different toolboxes for structural identifiability analysis.

	COMBOS	DAISY	EAR	GenSSI 1.0 and 2.0
Method	Differential algebra	Differential algebra	Semi-numerical probabilistic differential algebra	Generating series
Software base	Maxima	REDUCE	Mathematica	MATLAB
Availability	Web	Object Code	Object Code	Source Code
Operating system				
Windows		✓	✓	✓
Linux			✓	✓
Mac OS			✓	✓
Web	✓			
URL	http://biocyb1.cs.ucla.edu/combos	http://daisy.dei.unipd.it	http://www.fcc.chalmers.se/software/other-software/identifiabilityanalysis	https://github.com/genssi-developers/GenSSI
Reference	(Meshkat, et al., 2014)	(Bellu, et al., 2007)	(Anguelova, et al., 2012)	(Chiş, et al., 2011a)

Paper of first slide, supporting information

Tools for identifiability

Table S2. Features of structural identifiability toolboxes.

	COMBOS	DAISY	EAR	GenSSI 1.0	GenSSI 2.0
Model and experimental setup					
SBML import	×	×	×	×	✓
rational kinetic laws	×[1]	✓	✓	✓	✓
Parameter-dependent initial conditions	×[2]	×[2]	✓	✓	✓
Control inputs	✓	✓	✓	✓	✓
Multiple experimental conditions	×	×	×	×	✓
Transformation to polynomial form	×	×	×	×	✓
Identifiability analysis					
Local structural identifiability	✓	✓	✓	✓	✓
Global structural identifiability	✓	✓	×	✓	✓
Visualization of results	×	×	×	✓	✓
Post-processing					
State transformation	×	×	×	×	✓
Parameter transformation	×	×	×	×	✓

[1] COMBOS only supports polynomial kinetics, i.e. mass-action kinetics.

[2] COMBOS and DAISY do not support the usage of parameters that are not included in the vector field of the ODE model. Removing these parameters from our existing models changes them, making a comparison less meaningful.

Paper of first slide, supporting information

Tools for identifiability

https://arxiv.org/abs/1801.08112

Global Identifiability of Differential Models

Hoon Hong, Alexey Ovchinnikov, Gleb Pogudin, and Chee Yap

North Carolina State University, Department of Mathematics, Box 8205, Raleigh, NC 27695, USA
hong@ncsu.edu

CUNY Queens College, Department of Mathematics, 65-30 Kissena Blvd, Queens, NY 11367, USA
CUNY Graduate Center, Mathematics and Computer Science, 365 Fifth Avenue, New York, NY 10016, USA
aovchinnikov@qc.cuny.edu

Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA
pogudin@cims.nyu.edu, yap@cs.nyu.edu

mRNA Transfection

ELSEVIER

CrossMark

BASIC SCIENCE

Nanomedicine: Nanotechnology, Biology, and Medicine
10 (2014) 679–688

nanomedicine
Nanotechnology, Biology, and Medicine

Feature Article

nanomedjournal.com

Single-cell mRNA transfection studies: Delivery, kinetics and statistics by numbers

Carolin Leonhardt[a,1], Gerlinde Schwake[a,1], Tobias R. Stögbauer, PhD[a], Susanne Rappl[a], Jan-Timm Kuhr, PhD[b], Thomas S. Ligon, PhD[a], Joachim O. Rädler, PhD[a,*]

[a]*Faculty of Physics and Center for NanoScience (CeNS), Ludwig-Maximilians-Universität, München, Germany*
[b]*Institut für theoretische Physik, Technische Universität Berlin, Berlin-Charlottenburg, Germany*

Received 15 March 2013; accepted 18 November 2013

mRNA Transfection

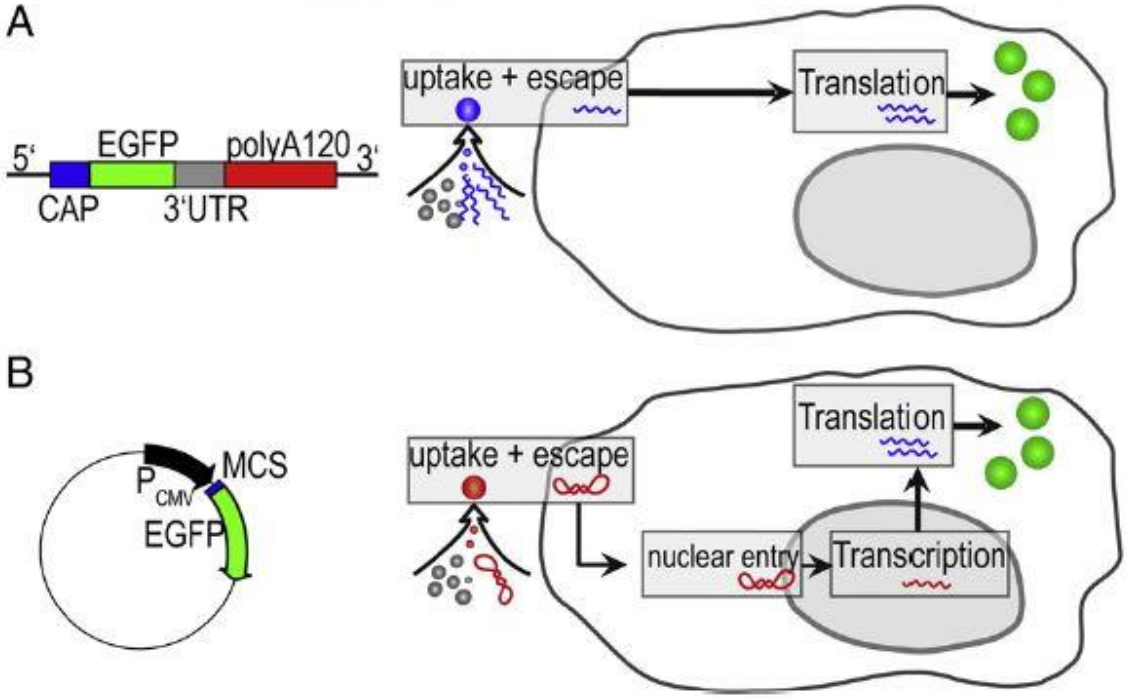


Figure 1. Comparison of mRNA and pDNA Vectors (both gene vectors encoding for the same eGFP protein) and their respective uptake pathways. (**A**) Linearized RNA (1192 bases) furnished with a stabilizing CAP sequence, an enhancing UTR sequence, and poly-(**A**) tail. (**B**) pDNA (4733 base pairs) under the control of the CMV promoter. The vector transfer under identical transfection protocols differs because mRNA is translated after endosomal escape, while plasmid DNA must be transferred into the nucleus for the initiation of transcription.

mRNA Transfection

mRNA transfection model

$$\frac{d}{dt}G = k_{TL} \cdot m - \beta \cdot G$$

$$\frac{d}{dt}m = -\delta \cdot m$$

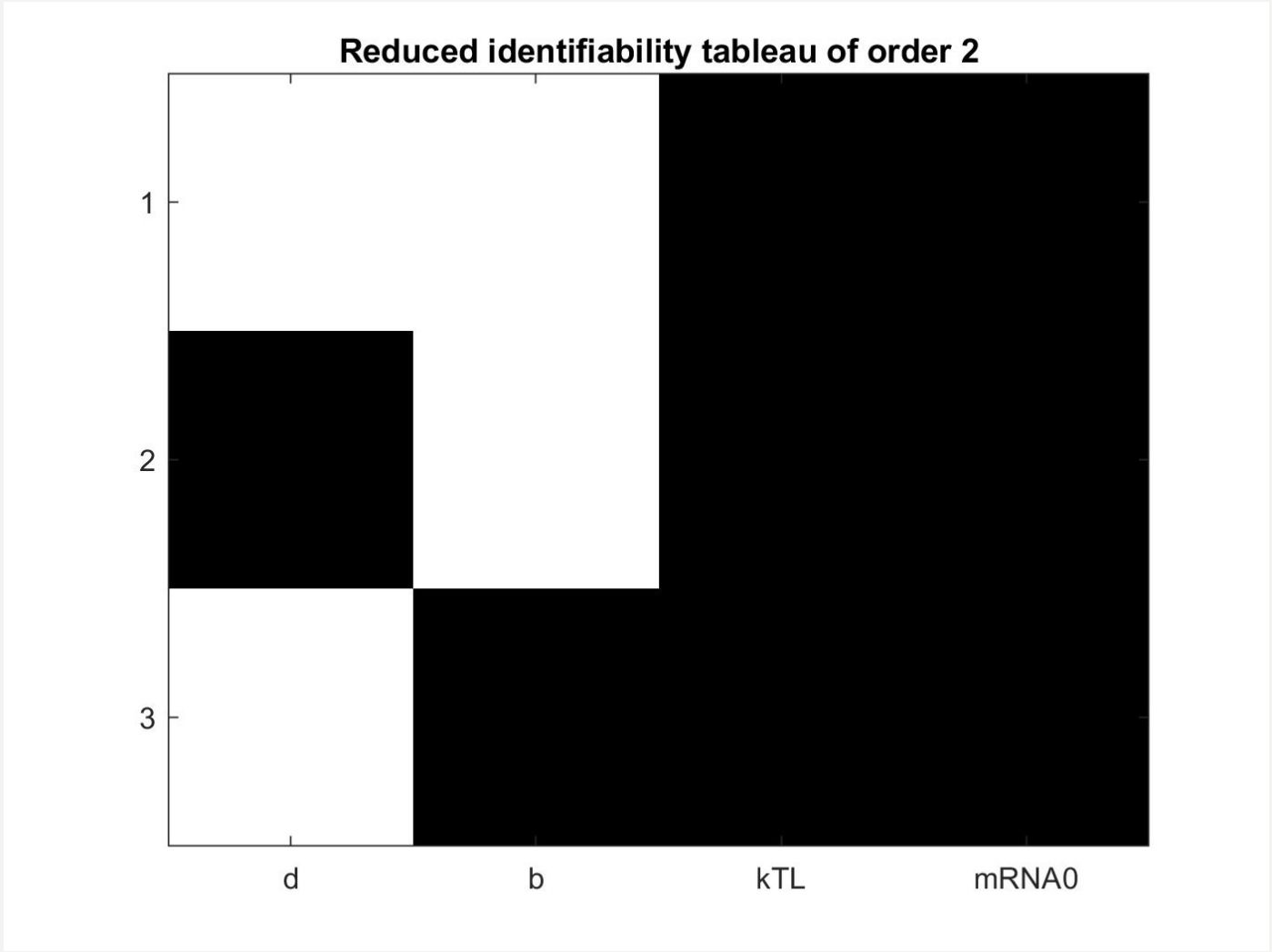
Solution

$$G_{mRNA}(t) = \frac{k_{TL} \cdot m_0}{\delta - \beta}\left(1 - e^{-(\delta-\beta)(t-t_0)}\right) \cdot e^{-\beta(t-t_0)}$$

Problems

- $k_{TL} \cdot m_0$ cannot be separated (identified).
- Solution is symmetric in β and δ.
- 2 equation for 4 parameters.

mRNA Transfection



GenSSI: Nothing is identifiable

```
--> WARNING: The number of parameters is Larger than the number of
   relations!
An explicit solution cannot be given for this subset of parameters.
   PLEASE CONSIDER AN EXTRA DERIVATIVE!

Structurally globally identifiable parameters:
[]
Structurally locally identifiable parameters:
[]
Structurally non-identifiable parameters:
[]
```

Experimental solutions

Measure not only GFP, but also mRNA

- …but that is difficult in living cells

Perform multiple experiments

- GFP with different degradation rate (eGFP)
- mRNA with different degradation rate (poly-A-tail)
- antibiotic that inhibits translation (different rate)

-> Multi-experiment models